



SPRING FRAMEWORK

BEAN NAMING & CONTEXT START-UP

APRIL, 2015

BEAN NAMING

`<bean id="..."` or `<bean name="..."`

- **id** - formal XML attribute, before Spring 3 allowed only restricted set of characters. Can have only single value.
- **name** - allows any characters, can have several values divided with comma/space/semicolon (all those values are aliases of a bean)

id and **name** can be present together

BEAN NAMING

ALIAS

```
<bean id="aaa" name="xxx,zzz" .. />
```

```
<alias name="aaa" alias="bbb"/>
```

SIMPLE VALUES

```
<bean id="client"  
      class="ua.epam.spring.core.beans.Client">  
  <constructor-arg value="1"/>  
  <constructor-arg value="John Smith"/>  
</bean>
```

WHEN ORDER MATTERS

```
<constructor-arg index="0" value="1"/>  
<constructor-arg index="1" value="J. Smith"/>
```

CONSTRUCTOR INJECTING

WHEN TYPE MATTERS

```
<constructor-arg type="java.lang.Integer"
    value="1"/>
<constructor-arg type="java.lang.String"
    value="1"/>
```

NAMES CAN BE USED

(Debug symbols are needed!)

```
<constructor-arg name="arg1" value="1"/>
<constructor-arg name="arg2" value="Smith"/>
```

OTHER BEANS

```
<bean id="app" class="ua.epam.spring.core.App"/>
```

```
    <constructor-arg ref="client"/>
```

```
    <constructor-arg ref="eventLogger"/>
```

```
</bean>
```

SPRING CONTAINER

- **BeanFactory**

Simple container that supports only DI

- **ApplicationContext**

Supports DI + framework services

- `ClassPathXmlApplicationContext`
- `FileSystemXmlApplicationContext`
- `AnnotationConfigApplicationContext`
- `XmlWebApplicationContext`
- `StaticApplicationContext`

```
public static void main(String[] args) {  
  
    ApplicationContext ctx =  
        new ClassPathXmlApplicationContext(  
            "spring.xml");  
  
    App app = (App) ctx.getBean("app");  
  
    app.logEvent("Some event for 1");  
    app.logEvent("Some event for 2");  
}
```




THANK YOU

YURIY_TKACH@EPAM.COM

APRIL, 2015